



Automatic Wrapper Generation and Data Extraction

Kristina Lerman
University of Southern California

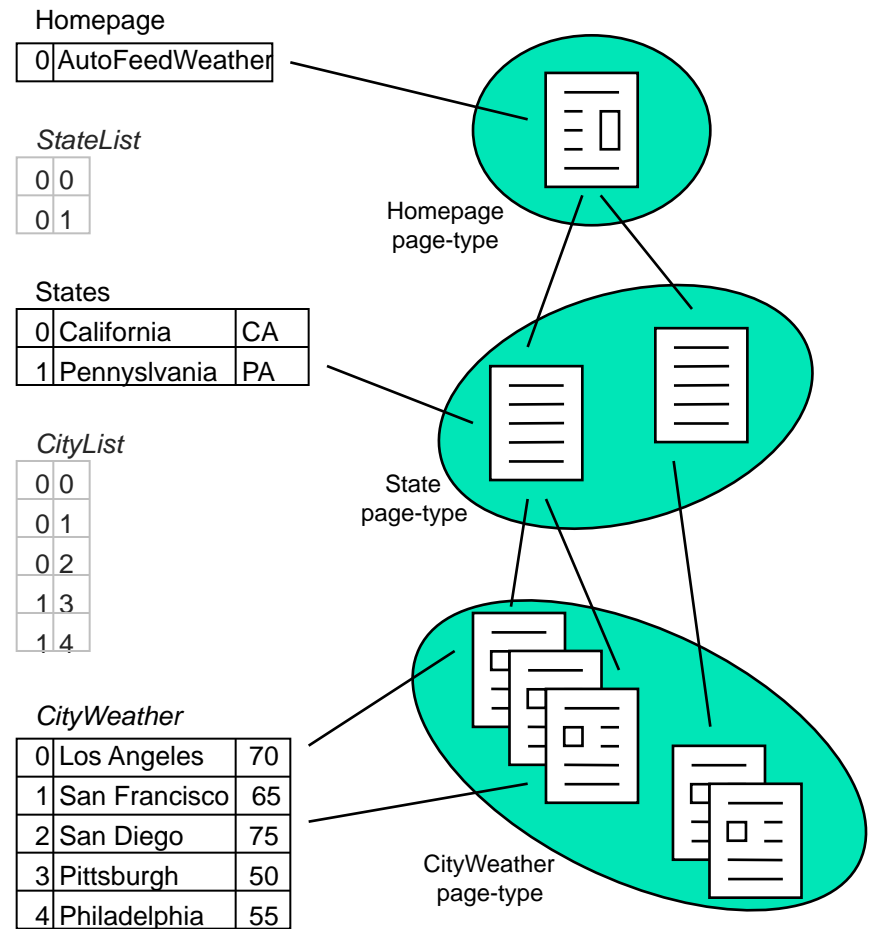


Overview

- Methods for automatic wrapper creation and data extraction
 - Grammar Induction approach
 - *Towards Automatic Data Extraction from Large Web Sites*
 - Website structure-based approach
 - *AutoFeed: An Unsupervised Learning System for Generating Webfeeds*
 - *Using the Structure of Web Sites for Automatic Segmentation of Tables*
 - Rule-based extraction from natural language text
 - *KnowItAll*
- No hand-labeled training examples are required!
 - Scaled to the size of the Web

Exploiting Structure of Web Sites

- Data-intensive Web sites present results in dynamically generated pages
- Web sites are highly *structured* in terms of
 - Organization of the site
 - Layout of pages
 - Content of data
- Exploit this structure for automatic information extraction

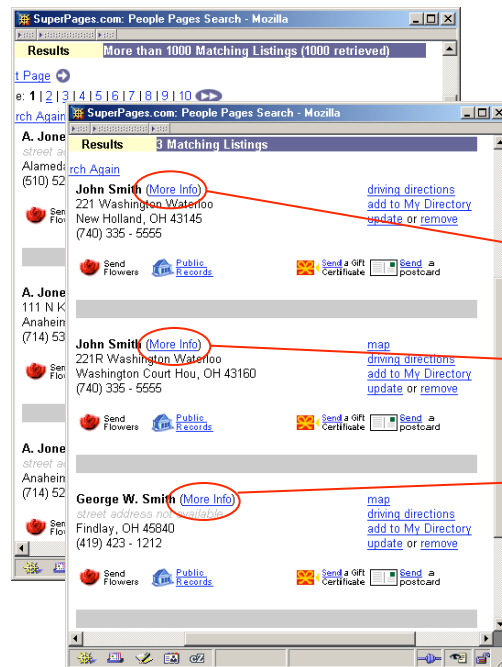




Using the Structure of Web Sites for Automatic Segmentation of Tables

Structure of Web Sites

Entry page → List pages → Detail pages



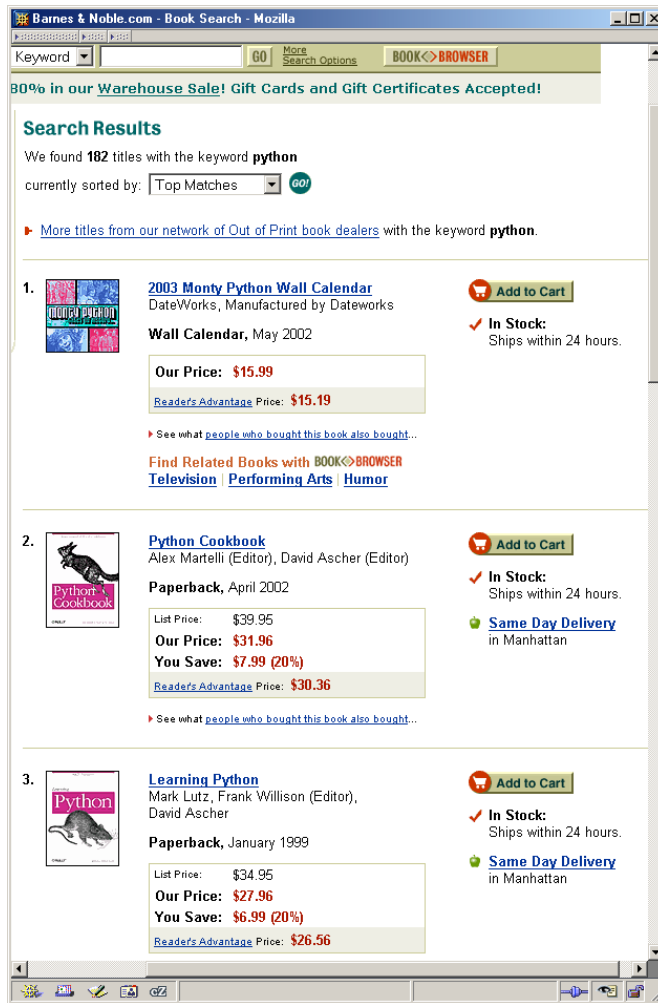
Structure of Pages and Data

The screenshot shows a web browser window displaying the SuperPages.com search results for 'John Smith'. The page features a navigation bar with categories like 'PEOPLE PAGES', 'CITY PAGES', and 'CONSUMER CENTER'. A sidebar on the left contains various utility links. The main content area displays three search results, each enclosed in a red oval. Each result includes the name, address, phone number, and several action links such as 'Send Flowers', 'Public Records', 'Send a Gift Certificate', and 'Send a Postcard'. The results are as follows:

Name	Address	Phone	Additional Info
John Smith	221 Washington Waterloo New Holland, OH 43145	(740) 335 - 5555	driving directions, add to My Directory, update or remove
John Smith	221R Washington Waterloo Washington Court Hou, OH 43160	(740) 335 - 5555	map, driving directions, add to My Directory, update or remove
George W. Smith	street address not available Findlay, OH 45840	(419) 423 - 1212	map, driving directions, add to My Directory, update or remove

- Data in the same “column” is of the same type
 - Each listing starts with NAME, followed by ADDRESS, CITY, STATE, etc.

Underlying Structure is not Always Clear



The screenshot shows a web browser window displaying search results for the keyword 'python' on the Barnes & Noble website. The page features a search bar at the top with the keyword 'python' and a 'GO' button. Below the search bar, there is a promotional banner for a 'Warehouse Sale' and a 'Search Results' section. The search results are sorted by 'Top Matches' and show 182 titles. Three results are visible:

- 2003 Monty Python Wall Calendar**
DateWorks, Manufactured by Dateworks
Wall Calendar, May 2002
Our Price: \$15.99
Reader's Advantage Price: \$15.19
In Stock: Ships within 24 hours.
- Python Cookbook**
Alex Martelli (Editor), David Ascher (Editor)
Paperback, April 2002
List Price: \$39.95
Our Price: \$31.96
You Save: \$7.99 (20%)
Reader's Advantage Price: \$30.36
In Stock: Ships within 24 hours.
Same Day Delivery in Manhattan
- Learning Python**
Mark Lutz, Frank Willison (Editor), David Ascher
Paperback, January 1999
List Price: \$34.95
Our Price: \$27.96
You Save: \$6.99 (20%)
Reader's Advantage Price: \$26.56
In Stock: Ships within 24 hours.
Same Day Delivery in Manhattan

- Variability of real-world data may obscure the underlying structure
 - Missing columns
 - “List Price” and “You save”
 - Formatting
 - Content



Problem Overview

Automatically, efficiently extract records from Web tables

Given a set of list and detail pages...

- Segment list data using information from detail pages
 - Logic based approach
 - Based on Constraint Satisfaction Problems (CSP)
 - Encode relations between data on list and detail pages as logical constraints and solve them
 - Probabilistic inference approach
 - Learns a model from data
 - Record segmentation is an assignment that maximizes the likelihood of data given the model

Identify Table and Extract Data



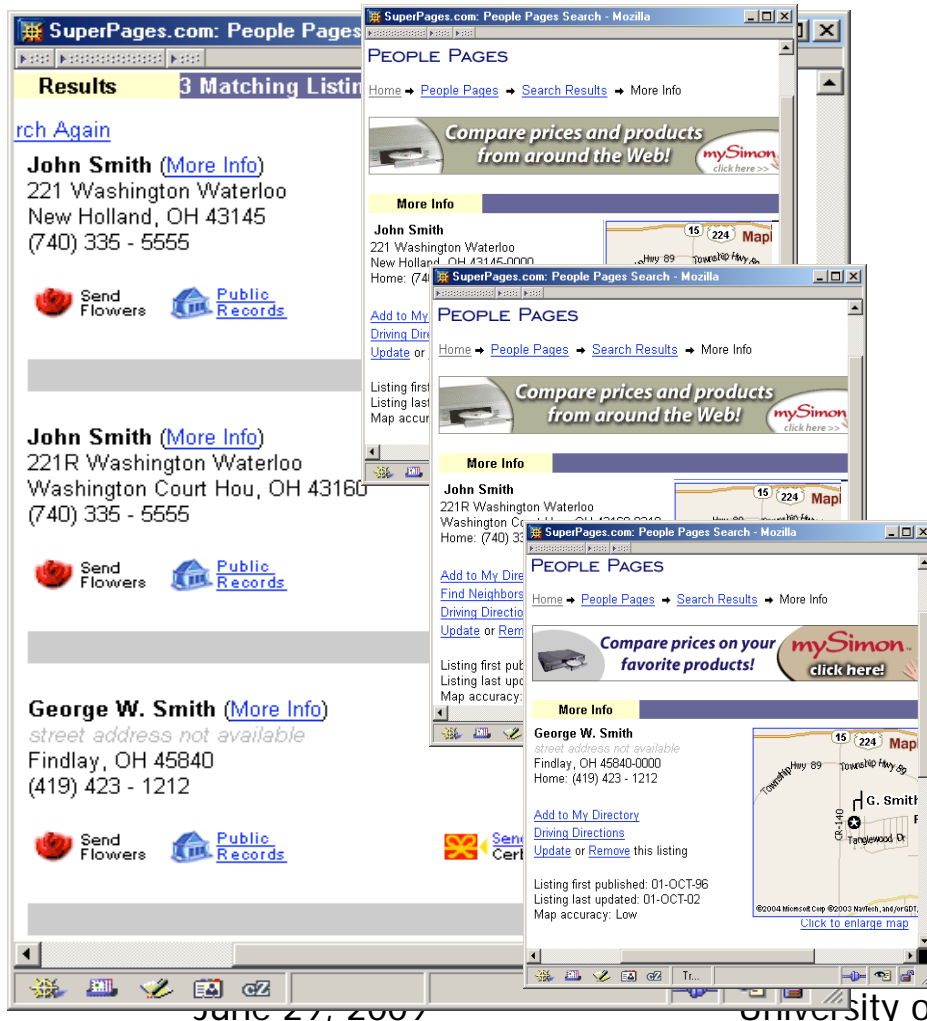
- Page template
 - Sequence of tokens shared by all pages
- Deduce page template
 - Given two or more example pages, derive the page template used to generate them
 - Table data and formatting tags are not part of the template
- Find table
 - Extract contiguous sequences of tokens from the largest page slot



Record Segmentation Basics (1)

- List and detail pages present two views of the same record
 - Some overlapping fields
- Each detail page is a distinct record
- Assumption: Web tables are laid out horizontally
 - Each record is in a separate row
 - Order in which extracts appear in the text stream of list page is the same order they appear in the table

Record Segmentation Basics (2)



For each extract E_j , record all detail pages on which it appears

- E_1 : John Smith r1, r2
- E_2 : 221 Was...terloo r1
- E_3 : New Hol...43145 r1
- E_4 : (740) 335-5555 r1, r2
- E_5 : John Smith r1, r2
- E_6 : 221R Was...erloo r2
- E_7 : Washingt...43160 r2
- E_8 : (740) 335-5555 r1, r2
- E_9 : George W. Smith r3
- E_{10} : Findlay, ... 45840 r3
- E_{11} : (419) 423-1212 r3

Record Segmentation Basics (3)

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}
r_1	1	1	1	1	1			1			
r_2	1			1	1	1	1	1			
r_3									1	1	1

- Observations of extracts on detail pages add valuable information for record segmentation
- Second record can be

- $E_4E_5E_6E_7E_8$

- $E_4E_5E_6E_7$

- $E_5E_6E_7E_8$

- $E_5E_6E_7$

- $E_6E_7E_8$

- E_6E_7



CSP Approach to Record Segmentation

- In CSP, problems are stated as logical expressions over variables
 - Pseudo-boolean (PB) representation
 - Variables are 0-1, constraints can be inequalities
 - Solution is assignment that minimizes inequality constraints
- Encode record segmentation problem in PB representation
 - Assignment variable x_{ij}
 - $x_{ij}=1$ when E_i is assigned to r_j
 - $x_{ij}=0$ when E_i is no part of r_j
 - Information from detail pages imposes constraints
 - Structure constraints
 - Position constraints

Structure Constraints

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}
r_1	1	1	1	1	1			1			
r_2	1			1	1	1	1	1			
r_3									1	1	1

- Uniqueness constraint
 - Every extract E_i belongs to exactly one record r_j

$$\sum_j x_{ij} = 1$$

- Consecutiveness constraint
 - Only contiguous blocks of extracts can be assigned to the same record



Structure Constraints

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}
r_1	1	1	1	1	1			1			
r_2	1			1	1	1	1	1			
r_3									1	1	1

- Uniqueness constraint
 - Every extract E_i belongs to exactly one record r_j
- Consecutiveness constraint
 - Only contiguous blocks of extracts can be assigned to the same record

$$x_{ij} + x_{kj} \leq 1 \text{ when there is } n, k < n < i, \text{ s.t. } x_{nj} = 0$$

Position Constraints

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}
p^{730}_1	1				1						
p^{772}_1		1									
p^{812}_1			1								
p^{846}_1				1				1			
p^{536}_2	1				1						
p^{578}_2				1				1			
p^{608}_2						1					
p^{642}_2							1				

- Position constraint
 - No two extracts assigned to same record can appear in the same position on the detail page
 - $pos_j(E_i) = pos_j(E_k)$, then E_i and E_k cannot be assigned to same record j
- Constraints are expressed mathematically and solved using integer optimization

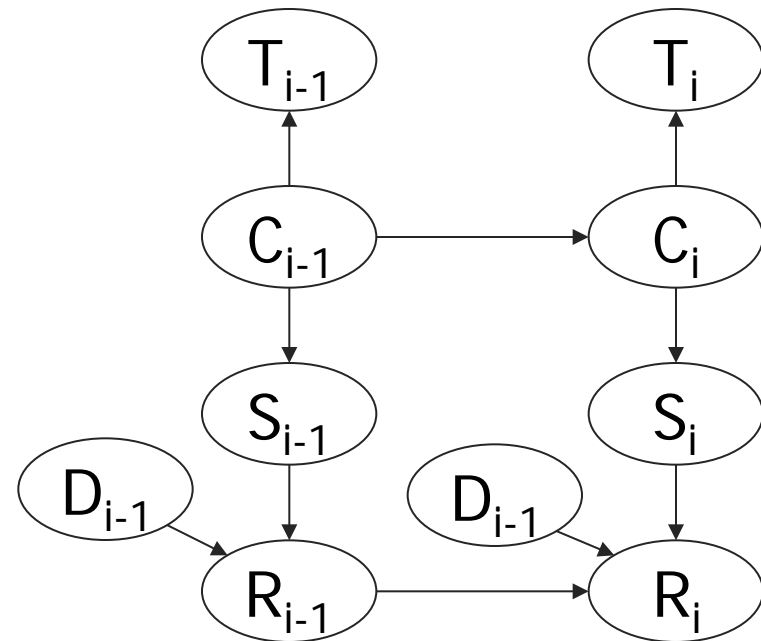


Probabilistic Approach to Record Segmentation

- Record segmentation as probabilistic inference
 - No labeled training examples
 - **Factor** the problem for efficient learning
 - **Bootstrap** the learning algorithm with information from detail pages
 - **Structure** constrain the problem further with global parameters such as record length

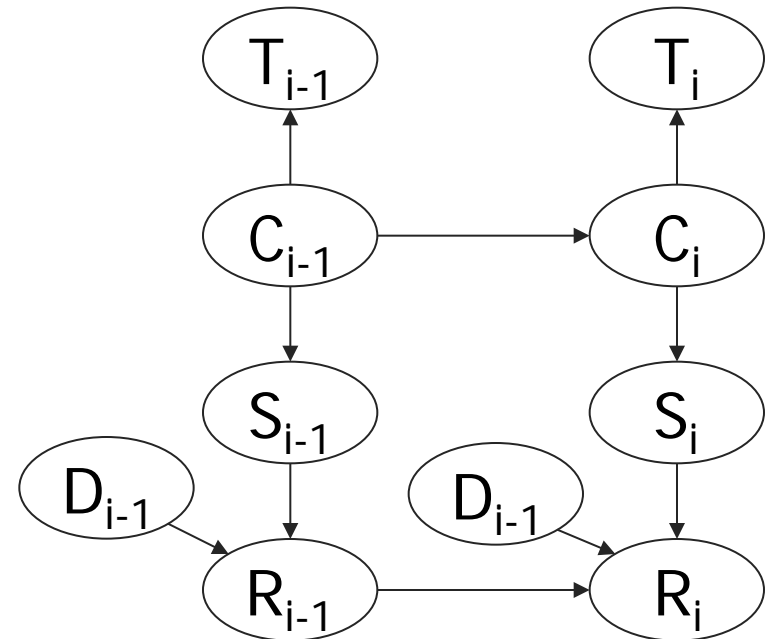
Probabilistic Model for Record Extraction: Variables

- Observed variables
 - $T = \{T_1 \dots T_n\}$ token types of extract E_i
 - $D = \{D_1 \dots D_n\}$ detail pages on which E_i was observed
- Unobserved variables
 - $R = \{R_1 \dots R_n\}$ record id
 - $C = \{C_1 \dots C_n\}$ column label
 - $S = \{S_1 \dots S_n\}$: $S_i = \text{true}$ if E_i is the start of a new record; false otherwise
- Dependencies
 - Given by arrows, eg, $P(C_i | C_{i-1})$
- Segmentation
 - find values for R and C given T, D variables: $\text{argmax } P(R, C | T, D)$



Probabilistic Model for Record Extraction: Dependencies

- $P(T_i|C_i)$: token type of E_i depends on column
- $P(C_i|C_{i-1})$: column label of E_i depends on previous column label (eg, *NAME* followed by *ADDRESS*, sometimes by *STATE*)
- $P(S_i|C_i)$: new record starts with a given column (eg, *NAME*)
- $P(R_i|R_{i-1}, D_i, S_i)$: record number of E_i depends on record number of previous extract, whether it starts a new record, and detail pages on which it was observed.





Learning the Model

- Constrain the problem further
 - Bootstrap
 - Detail pages provide initial guesses for parameters
 - $P(R_i=r_i)$
 - Evidence about where records start: $P(S_i=true)=1$
 - Token types of columns $P(T_j|C_i)$
 - Structure
 - Table has π columns specified by the underlying database schema
 - However, not every record will have an attribute for every field, i.e., not every record has π fields
 - Number of fields in a record estimated from data

Learning the Model

Initial guess for record assignment $P(R_i)$

$P(R_i=r_j)$	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}
r_1	1/2	1	1	1/2	1/2			1/2			
r_2	1/2			1/2	1/2	1	1	1/2			
r_3									1	1	1

Initial guess for record start $P(S_i)$

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}
$P(S_i)$	1	1		1	1	1			1		

Initial guess for length of records

π_k	2	3	4	5	6	7	...				
$P(\pi_k)$	2/14	6/14	4/14	2/14	0	0	0				



Learning Algorithm

- Use EM to implement the inference algorithm
 1. Initial guess for π_k for each record j
 2. For each potential record, update $P(C_i | T_i, C_{i-1})$
 3. Update $P(S_i | C_i)$
 4. Update $P(R_i | R_{i-1}, D_i, S_i)$

Result is the most likely assignment of data to R
and C = record segmentation



Validation

- Input data
 - list and detail pages from 12 sites in domains: *book sellers, property tax, white pages, corrections*
- Metrics
 - $P = Cor / (Cor + InCor + NonRecords)$
 - $R = Cor / (Cor + UnsegRecords)$
 - $F = 2PR / (P + R)$
- Results
 - CSP approach: $P=0.85, R=0.84, F=0.84$
 - Probabilistic approach: $P=0.74, R=0.99, F=0.85$
 - Good performance for an automatic algorithm!

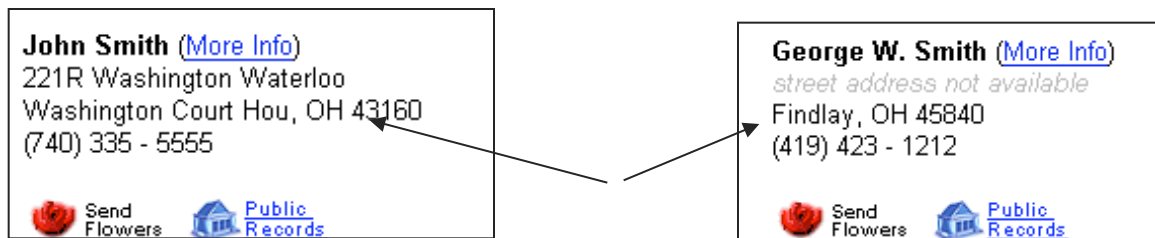


Discussion of Results

- CSP approach is very reliable on clean data, but sensitive to errors in data source
 - Attribute has one value on list page and another on detail page
- Probabilistic approach tolerates inconsistencies and is more expressive
- Combination of two techniques may be more robust

Comparison with RoadRunner

- RoadRunner System
 - Automatically learns the page and table template by exploiting similarities in page layout (HTML tags)
 - Uses the template to automatically extract data
 - Does not allow for disjunctions
 - Disjunctions are necessary to represent alternative layout instructions for the same field





Discussion

- Domain-independent approach for automatically extracting and segmenting data from Web tables
- Approach leverages additional information provided by Web site structure
 - Logic based approach
 - Information provided by detail pages encoded as constraints and solved to obtain record segmentation
 - Probabilistic inference approach
 - Information provided by detail pages and table structure represented as a probabilistic model
 - Use inference to learn proper segmentation
- Validated approach on 12 Web sites from diverse information domains
 - Efficient, accurate performance, $F=0.85$ and $F=0.84$



KnowItAll: Methods for Domain-Independent Information Extraction from the Web



Automatic Data Extraction

- Extract data from the Web without hand-labeled training examples
- Types of information extracted
 - Data tuples
 - '4676 Admiralty Way', 'Marina del Rey', 'CA', '90292', 'USA'
 - Facts
 - Entities
 - 'Los Angeles', 'Albert Einstein'
 - Classes and relations
 - Class instances:
 - 'Los Angeles' is a CITY
 - 'Albert Einstein' is a SCIENTIST



KnowItAll Approach

Two-stage approach to automatic data extraction

- Extraction patterns to generate candidate facts
 - Pattern “NP1 *such as* NP2”
“... tours in **cities *such as* Paris and Berlin**”
 - Extracts class *CITY* with instances *Paris* and *Berlin*
- Text candidate facts using Pointwise Mutual Information (PMI)
 - Statistics computed from all text on Web
 - Use existing Web search technology to efficiently compute statistics
 - Associates a probability with every fact it extracts
 - Automatically manage tradeoff between precision and recall



Extractor

- Extractor –natural language patterns to extract instances of classes
NP1 "such as" NPList2
& head(NP1)= plural(Class1)
& properNoun(head(each(NPList2)))
=> instanceOf(Class1, head(each(NPList2)))
keywords: "plural(Class1) such as"
- Uses part-of-speech tagger to identify Noun Phrases (NP)



Search Engine Interface

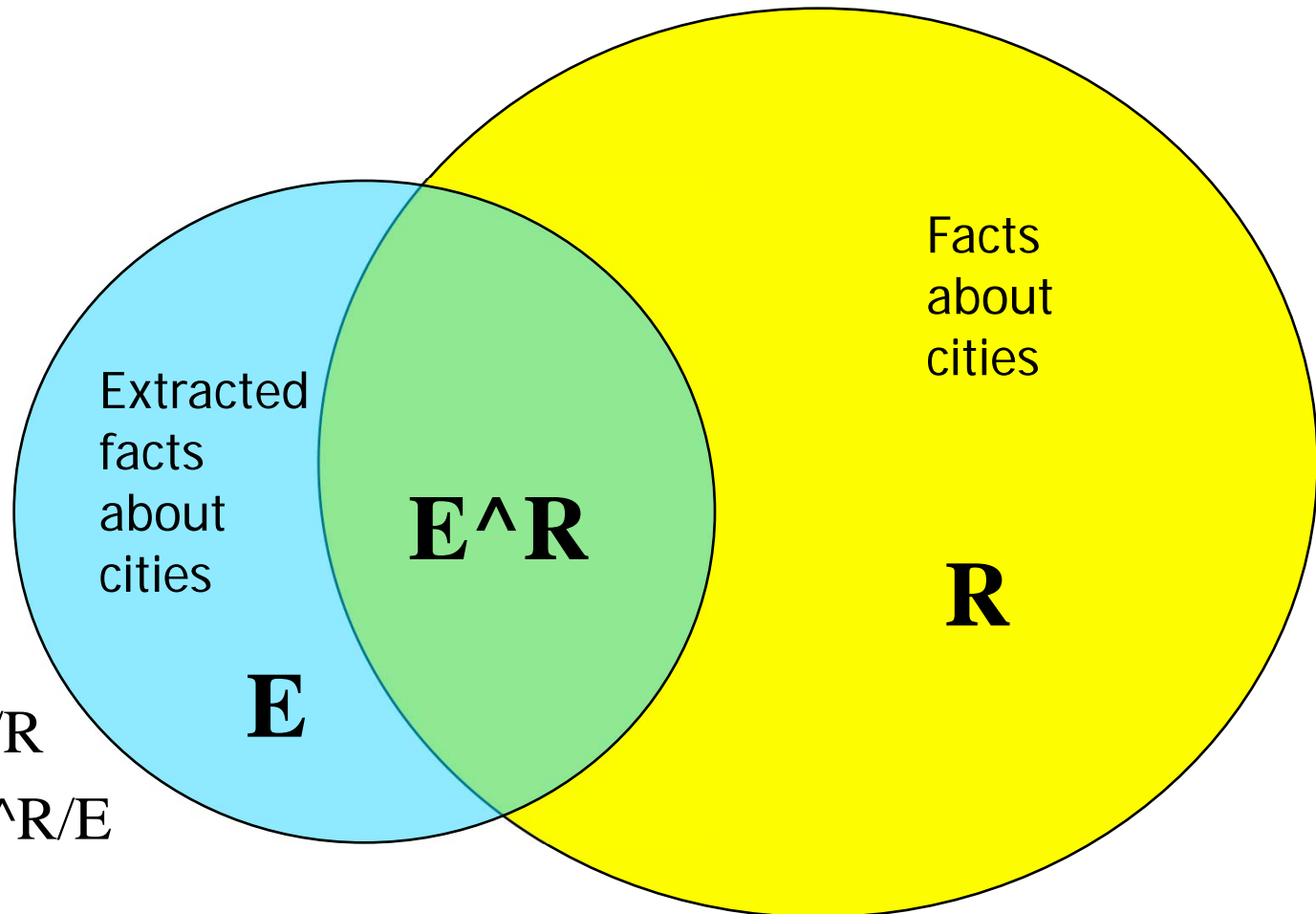
- Query search engine with phrases
 - “cities such as”
- Apply Extractor to all pages return by search



Assessor

- Uses statistics computed over all Web pages to assess the likelihood that extracted fact I is correct
 - Pointwise Mutual Information (PMI)
$$\text{PMI}(I, D) = \frac{|\text{Hits}(D+I)|}{|\text{Hits}(I)|}$$
 - D is discriminator phrase: e.g., “city of”
 - $\text{Hits}(x)$ = number of Web pages that contain x
 - PMI is a feature to Naïve Bayes Classifier
 - More likely classes get higher probabilities
 - Probability threshold tunable parameter to increase precision (at expense of recall)

Precision and Recall – a recap



$$\text{Recall} = E^R/R$$

$$\text{Precision} = E^R/E$$

Goal: Make the blue circle overlap more of the yellow circle!



Enhancements to KnowItAll

- Enhancements to increase precision & recall
 - Rule Learning
 - Learns domain specific rules and validates accuracy of instances they extract
 - Subclass Extraction
 - Automatically identify subclasses
 - Learn that physicists, geologists, etc. are subclasses of scientists
 - Rule “physicists such as ...” will extract more scientists
 - List Extraction
 - Locate lists of class instances
 - Learns a wrapper for the list to extract instances



Enhancements: Rule Learning

- Learn domain-specific rules to increase KnowItAll's precision and recall
E.g., "... headquartered in <CITY> ..."
 1. Start with instances extracted by generic patterns
 2. Query search engine with instances → pages
 3. From each page, extract *context string* for instance
 - 4 words before, and after
 4. 'Best' substrings of the 'best' context strings are converted to new Extraction Rules that extract new instances with high precision
 - Heuristic: Prefer substrings that appear in multiple pages
 - Heuristic: Penalize substrings that lead to many false positives



Examples of Rule Learning

Most productive rules learn for each class, with number of correct extractions and precision

1.	the cities of <i><city></i>	5215	0.80
2.	headquartered in <i><city></i>	4837	0.79
3.	for the city of <i><city></i>	3138	0.79
4.	in the movie <i><film></i>	1841	0.61
5.	<i><film></i> the movie starring	957	0.64
6.	movie review of <i><film></i>	860	0.64
7.	and physicist <i><scientist></i>	89	0.61
8.	physicist <i><scientist></i> ,	87	0.59
9.	<i><scientist></i> , a British scientist	77	0.65



Subclass Extraction

- Identify subclasses and instantiate new generic patterns
 - PHYSICIST is a subclass of SCIENTIST → new rule “physicists such as ...”
 - Increases KnowItAll coverage
- Subclasses of SCIENTIST found by KnowItAll
 - biologist
 - astronomer
 - mathematician
 - geologist
 - chemist
 - anthropologist
 - psychologist
 - paleontologist
 - engineer
 - zoologist
 - meteorologist
 - economist
 - sociologist
 - oceanographer
 - pharmacist
 - climatologist
 - neuropsychologist
 - microbiologist



Subclass Extraction

1. Apply Subclass Extraction rules to extract candidate subclasses
 - "... such C_1 as CN ..." \rightarrow CN is subclass of C_1 .
 - "... CN and other C_1 ..." \rightarrow CN is subclass C_1 .
2. Assess validity of candidate
 - Is subclass in a reference taxonomy (WordNet)?
 - Check word morphology \rightarrow "*microbiologist*" is a subclass of "biologist"



List Extraction

- Extract information from formatted lists
 - Approach
 - Query search engine with k random instances extracted by KnowItAll
 - In each Web page, search for a list containing these keywords using HLRT-like wrapper induction algorithm*
 - Convert Web page to DOM tree
 - Select subtrees corresponding to positive examples
 - Finds greatest common prefixes (and suffixes) for these examples
 - Choose header and tail strings to limit extraction to good subtrees
- *Can learn wrapper from few positive examples
- Assess the likelihood of each extracted instance
 - Rank instances by the number of lists they appear in



Evaluation

- For classes CITY and FILM
 - Extracted >40k (compared to baseline 10k) at 90% precision
 - Most of the improvement due to List Extraction
- For class SCIENTIST
 - Extracted 40k instances (compared to baseline ~2k) at 90% precision
 - Most of the improvement due to Subclass Extraction



Discussion

- Automatic collection of large body of facts
- Extract facts (e.g., instances of classes) from text using generic NLP rules
- Heuristics added to KnowItAll (rule learning, subclass extraction, list extraction) greatly improve recall while maintaining high precision



Conclusion

- Covered method to automatically extract massive data sets from Web pages
 - Structured pages
 - Natural language text
- Extraction from structured Web pages
 - Exploit structure in pages (grammar)
 - Exploit structure of site
- Extraction from text
 - Exploit NLP rules and Web statistics to extract high quality facts